Contents

TCP Traffic Shifting

4 minute read ✔ page test

This task shows you how to shift TCP traffic from one version of a microservice to another.

A common use case is to migrate TCP traffic gradually from an older version of a microservice to a new one. In Istio, you accomplish this goal by configuring a sequence of routing rules that redirect

a percentage of TCP traffic from one destination to another.

In this task, you will send 100% of the TCP traffic to `tcp-echo:v1`. Then, you will route 20% of the TCP traffic to `tcp-echo:v2` using Istio's weighted routing feature.

Before you begin

- Setup Istio by following the instructions in the

Installation guide.

- Review the Traffic Management concepts doc.

Set up the test environment

1. To get started, create a namespace for testing TCP traffic shifting and label it to enable automatic sidecar injection.

```
$ kubectl create namespace istio-io-tcp-traffic-shifting
$ kubectl label namespace istio-io-tcp-traffic-shifting ist
io-injection=enabled
```

2. Deploy the `sleep` sample app to use as a test source for sending requests.

```
$ kubectl apply -f @samples/sleep/sleep.yaml@ -n istio-io-tcp-traffic-shifting
```

3. Deploy the `v1` and `v2` versions of the `tcp-echo` microservice.

```
$ kubectl apply -f @samples/tcp-echo/tcp-echo-services.yaml@ -n istio-io-tcp-traffic-shifting
```

4. Follow the instructions in Determining the ingress IP and ports to define the `TCP_INGRESS_PORT` and `INGRESS_HOST` environment variables.

Apply weight-based TCP routing

1. Route all TCP traffic to the `v1` version of the `tcp-echo` microservice.

   ```
   $ kubectl apply -f @samples/tcp-echo/tcp-echo-all-v1.yaml@
   -n istio-io-tcp-traffic-shifting
   ```

2. Confirm that the `tcp-echo` service is up and running by sending some TCP traffic from the `sleep` client.

```
$ for i in {1..20}; do \
kubectl exec "$(kubectl get pod -l app=sleep -n istio-io-tc
p-traffic-shifting -o jsonpath={.items..metadata.name})" \
-c sleep -n istio-io-tcp-traffic-shifting -- sh -c "(date;
sleep 1) | nc $INGRESS_HOST $TCP_INGRESS_PORT"; \
done
one Mon Nov 12 23:24:57 UTC 2018
one Mon Nov 12 23:25:00 UTC 2018
one Mon Nov 12 23:25:02 UTC 2018
one Mon Nov 12 23:25:05 UTC 2018
one Mon Nov 12 23:25:07 UTC 2018
one Mon Nov 12 23:25:10 UTC 2018
one Mon Nov 12 23:25:12 UTC 2018
one Mon Nov 12 23:25:15 UTC 2018
one Mon Nov 12 23:25:17 UTC 2018
one Mon Nov 12 23:25:19 UTC 2018
...
```

You should notice that all the timestamps have a

prefix of *one*, which means that all traffic was routed to the `v1` version of the `tcp-echo` service.

3. Transfer 20% of the traffic from `tcp-echo:v1` to `tcp-echo:v2` with the following command:

```
$ kubectl apply -f @samples/tcp-echo/tcp-echo-20-v2.yaml@ -n istio-io-tcp-traffic-shifting
```

   Wait a few seconds for the new rules to propagate.

4. Confirm that the rule was replaced:

```
$ kubectl get virtualservice tcp-echo -o yaml -n istio-io-tcp-traffic-shifting
apiVersion: networking.istio.io/v1beta1
```

```
kind: VirtualService
  ...
spec:
  ...
  tcp:
  - match:
    - port: 31400
    route:
    - destination:
        host: tcp-echo
        port:
          number: 9000
        subset: v1
      weight: 80
    - destination:
        host: tcp-echo
        port:
          number: 9000
        subset: v2
      weight: 20
```

5. Send some more TCP traffic to the `tcp-echo` microservice.

```
$ for i in {1..20}; do \
kubectl exec "$(kubectl get pod -l app=sleep -n istio-io-tc
p-traffic-shifting -o jsonpath={.items..metadata.name})" \
-c sleep -n istio-io-tcp-traffic-shifting -- sh -c "(date;
sleep 1) | nc $INGRESS_HOST $TCP_INGRESS_PORT"; \
done
one Mon Nov 12 23:38:45 UTC 2018
two Mon Nov 12 23:38:47 UTC 2018
one Mon Nov 12 23:38:50 UTC 2018
one Mon Nov 12 23:38:52 UTC 2018
one Mon Nov 12 23:38:55 UTC 2018
two Mon Nov 12 23:38:57 UTC 2018
one Mon Nov 12 23:39:00 UTC 2018
one Mon Nov 12 23:39:02 UTC 2018
one Mon Nov 12 23:39:05 UTC 2018
one Mon Nov 12 23:39:07 UTC 2018
...
```

You should now notice that about 20% of the

timestamps have a prefix of *two*, which means that 80% of the TCP traffic was routed to the `v1` version of the `tcp-echo` service, while 20% was routed to `v2`.

Understanding what happened

In this task you partially migrated TCP traffic from an old to new version of the `tcp-echo` service using Istio's weighted routing feature. Note that this is very

different than doing version migration using the deployment features of container orchestration platforms, which use instance scaling to manage the traffic.

With Istio, you can allow the two versions of the `tcp-echo` service to scale up and down independently, without affecting the traffic distribution between them.

For more information about version routing with autoscaling, check out the blog article Canary Deployments using Istio.

Cleanup

1. Remove the `sleep` sample, `tcp-echo` application, and routing rules:

```
$ kubectl delete -f @samples/tcp-echo/tcp-echo-all-v1.yaml@
 -n istio-io-tcp-traffic-shifting
$ kubectl delete -f @samples/tcp-echo/tcp-echo-services.yam
l@ -n istio-io-tcp-traffic-shifting
$ kubectl delete -f @samples/sleep/sleep.yaml@ -n istio-io-
tcp-traffic-shifting
$ kubectl delete namespace istio-io-tcp-traffic-shifting
```